ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Laboratory

TEC, NSG, DISCO

HS 2020 Prof. L. Vanbever, Prof. R. Wattenhofer, Prof. L. Thiele

Discrete Event Systems

Exam

Tuesday, 9th February 2021, 09:00–11:00.

Do not open or turn before the exam starts!
Read the following instructions!

The exam takes 120 minutes and there is a total of 120 points. The maximum number of points for each subtask is indicated in brackets. **Justify all your answers** unless the task explicitly states otherwise. Mark drawings precisely.

Answers which we cannot read are not awarded any points!

At the beginning, fill in your name and student number in the corresponding fields below. You should fill in your answers in the spaces provided on the exam. If you need more space, we will provide extra paper for this. Please label each extra sheet with your name and student number.

Name	Legi-Nr.

Points

Question	Topic	Achieved Points	Maximal Points
1	True or False		5
2	Regular Languages		15
3	Context-Free Languages		20
4	Secret Message Passing		20
5	Elevator		20
6	True or False		6
7	Binary Decision Diagram		10
8	Petri Nets		24
Total			**120**

1 True or False (5 points)

On the following page you see definitions of languages L_1, ..., L_3. Based on those, indicate for each of the statements a)–e) whether they are **TRUE** or **FALSE** (circle **one** answer) and give a brief explanation. For each question answered correctly, one point is added. For each question answered incorrectly, one point is removed. No answer gives zero points. There is always one correct answer. This task gives at least 0 points.

a) [1] **TRUE** **FALSE** | Let $\Sigma = \{a\}$. Then, $L = \{a^{x+y} \mid x \in L_1, y \in L_2\}$ is regular.

b) [1] **TRUE** **FALSE** | $L = \left\{0^n 1^n \;\middle|\; n < 10^{10^{10}}\right\}$ is not regular.

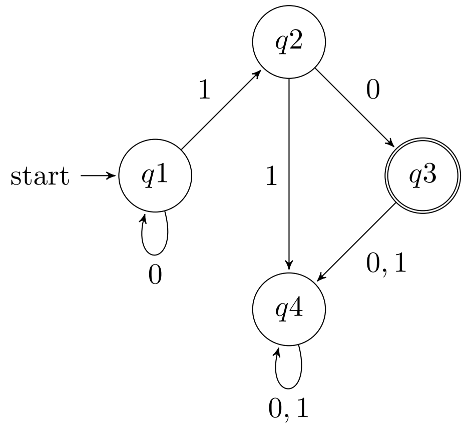
c) [1] **TRUE** **FALSE** | L_3 can be written as the REX: $0^*1\left((01)^* \cup 0^*\right)(1 \cup 0)$.

d) [1] **TRUE** **FALSE** | $L_2 \subseteq L_3$.

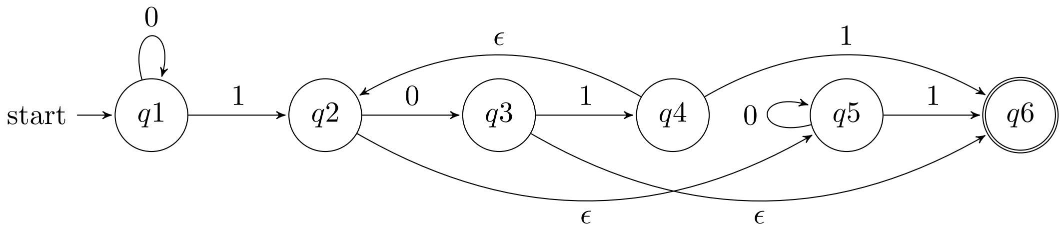
e) [1] **TRUE** **FALSE** | $L = \{uvv \mid u \in L_3, v \in L_2\}$ is context-free.

The languages L_1, ..., L_3 are defined over the alphabet $\Sigma = \{0, 1\}$ as follows:

- $L_1 = \{01, 10, 11\}$,
- L_2 is the language recognized by the following DFA:



- L_3 is the language recognized by the following NFA:



Model solution

a) True. L_1 is finite and L_2 is arithmetically bounded to a single value. In other words, $L = \{aaa, aaaa, aaaaa\}$.

b) False. L is finite and thus regular.

c) False. Counter examples (one direction is sufficient):

- $100 \notin L_3$ is accepted by the REX.
- $010101 \in L_3$, but not possible with the REX.

$$q1 \xrightarrow{0} q1 \xrightarrow{1} q2 \xrightarrow{0} q3 \xrightarrow{1} q4 \xrightarrow{\varepsilon} q2 \xrightarrow{\varepsilon} q5 \xrightarrow{0} q5 \xrightarrow{1} q6$$

d) True. $L_2 = 0^*10$ is a subset of L_3:

$$q1 \left(\xrightarrow{0} q1 \right)^* \xrightarrow{1} q2 \xrightarrow{0} q3 \xrightarrow{\varepsilon} q6$$

e) True. $S \to L_3 V 10$, $V \to 0V0 \mid 10$, where L_3 produces all words in L_3.

2 Regular Languages (15 points)

a) Consider the generic language L_n for $n \geqslant 2$ over the alphabet $\Sigma_n = \{x_1, x_2, ..., x_n\}$:

$$L_n = \Big\{ w \ \mid \ \#_{x_1}(w) \ \mathsf{mod}\ k \ > \ \#_{x_2}(w) \ \mathsf{mod}\ k \ > \ \dots \ > \ \#_{x_n}(w) \ \mathsf{mod}\ k \Big\}$$

where $k = n + 1$. For example, $L_2 = \{w \mid \#_{x_1}(w) \ \mathsf{mod}\ 3 > \#_{x_2}(w) \ \mathsf{mod}\ 3\}$.

Recall: $\#_{x_i}(w)$ *denotes the number of occurrences of the symbol* $x_i \in \Sigma$ *in a word* $w \in \Sigma^*$.

(i) [6] Draw a DFA recognizing L_2. Use at most 12 states.

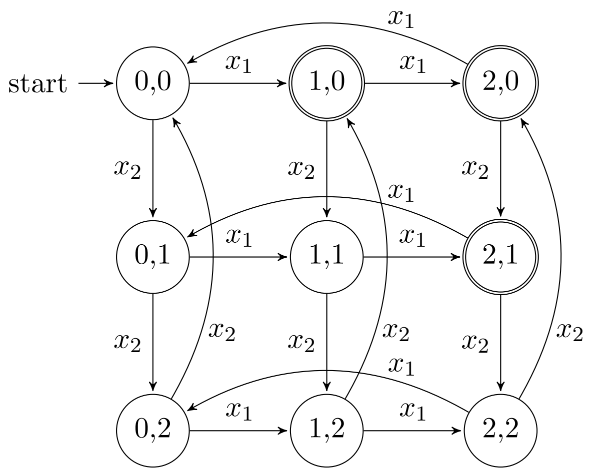
(ii) [2] Give the size of the minimized DFA for L_n as a function of n.

(iii) [2] Give the number of final states in the minimized DFA for L_n as a function of n.

b) [5] Consider $L = \{x\#y\#z|\ x, y, z \in \{0,1\}^*, z - y = x\}$ where x, y and z are unsigned binary numbers. State whether L is regular or not, and prove your claim.

Model solution

a) (i) A DFA for L_2 looks as follows:



(ii) $size(n) = \underbrace{k \cdot k \cdot \ldots \cdot k}_{n \text{ times}} = (n+1)^n$.

(iii) For $n \geqslant 2$, there are always exactly $n+1$ final states.

This can be seen inductively. The claim holds true for $n = 2$ as we can see above. For the induction step $n \to n+1$, there is one more dimension to consider, and we have one more available number to fit it in the >-relation. Hence, for L_{n+1} we get all the accepting state combinations for L_n in state n of the new dimension ($= n+1$), plus one configuration that never needed the highest number in state $n-1$ of the newly added dimension.

b) We claim that L is not regular and prove our claim with the pumping lemma.

1. Assume for contradiction that L was regular.
2. There must exist some p, s.t. any word $w \in L$ with $|w| \geqslant p$ is pumpable.
3. Choose the string $w = 1^p\#0\#1^p \in L$ with length $|w| > p$.
4. Consider all ways to split $w = xyz$ s.t. $|xy| \leqslant p$ and $|y| \geqslant 1$.
   $\to$ Hence, $y \in 1^+$.
5. Observe that $xy^0z \notin L$ – a contradiction to p being a valid pumping length.
6. Consequently, L cannot be regular. $\square$

3 Context-Free Languages (20 points)

a) Consider $L = \{x \mid x \textit{ is divisible by 5}\}$ where x is an unsigned binary number.

(i) `[6]` Give a CFG for L. Use at most 7 non-terminal symbols.

(ii) `[2]` Give derivations of $x_1 = 0101$ and $x_2 = 1111$ with your CFG.

(iii) `[2]` Is your CFG ambiguous?

b) `[10]` Draw a PDA that recognizes $L = \{x\#y \mid x, y \in \{0,1\}^+,\ x \neq y\}$. Use at most 12 states.

Hint: Can two strings x and y with $|x| < |y|$ ever be equal?

Model solution

a) For the CFG, we first accept all leading 0s and then have a symbol S_i that allows to derive all bit-strings with a remainder of $S_i \mod 5 = i$.

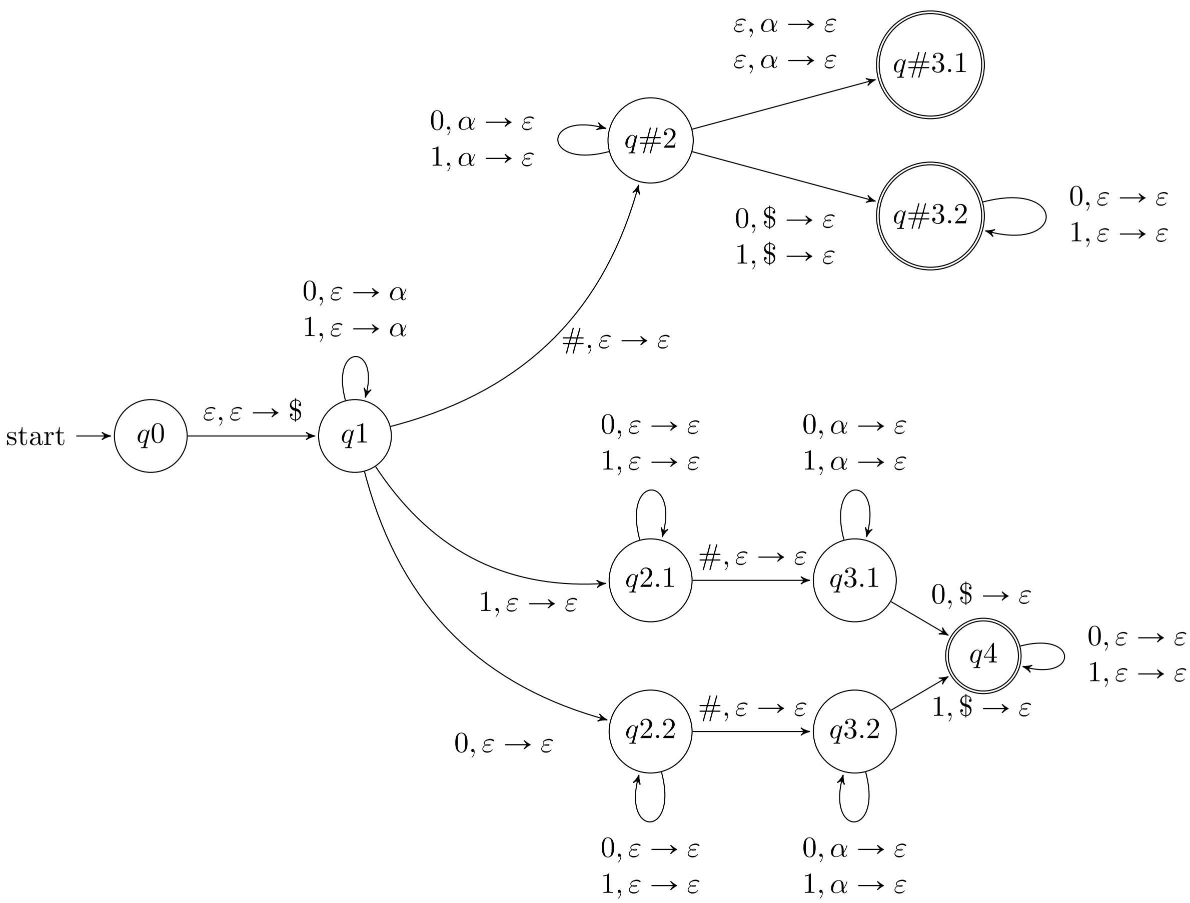
(i)

$$\begin{aligned} S &\to 0S \mid S_0 \\ S_0 &\to S_00 \mid S_21 \mid \varepsilon \\ S_1 &\to S_30 \mid S_01 \\ S_2 &\to S_10 \mid S_31 \\ S_3 &\to S_40 \mid S_11 \\ S_4 &\to S_20 \mid S_41 \end{aligned}$$

(ii) $0101 : S \to 0S \to 0S_0 \to 0S_21 \to 0S_101 \to 0S_0101 \to 0101$
$1111 : S \to S_0 \to S_21 \to S_311 \to S_1111 \to S_01111 \to 1111$

(iii) The grammar given is ambiguous: For leading zeros, there are currently 2 ways to generate them. Either via $S \to 0S$, or via $S_0 \to S_00$ (see the second derivation for an example how S_0 will end up in the front). All other substrings have a unique derivation, as their remainder $S_i \mod 5 = i$ is unique.
The grammar could be made non-ambiguous by removing the rule $S \to 0S \mid S_0$ and making S_0 the start symbol.

b) A PDA recognizing $L = \{x\#y \mid x, y \in \{0,1\}^+,\ x \neq y\}$ could look like this:



We make use of non-determinism for two decisions: First, we decide whether to check that the lengths $|x|$ and $|y|$ differ, i.e. whether $|x| > |y|$ ($\rightarrow q\#3.1$) or $|x| < |y|$ ($\rightarrow q\#3.2$). If they do not differ by length, we use non-determinism to fix some position in x that is a 1 (or a 0), and then we make sure that the same digit in y is a 0 (or a 1; $\rightarrow q4$). As both $x, y \in \{0,1\}^+$ and have the same length, there must always exist such a position.

Note that the presented automaton is not minimal, as states $q4$ and $q\#3.2$ are indistinguishable. However, we leave the presentation "as is" for clarity of the construction idea.

4 Secret Message Passing (20 points)

A group of 9 students decides to use a diary to start passing messages around. The students take turns. A student is keeping the diary for a day (possibly writing something in the diary), then randomly passing it to one of their friends. These friend relations are depicted in Figure 1 (note that a friend relation is always mutual).

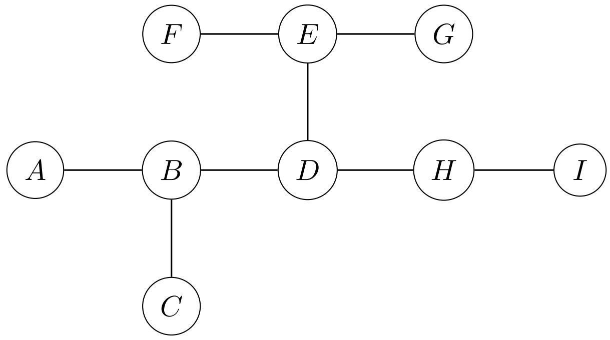


Figure 1: Friend Relations

a) [3] Assume Alice (A) has the diary on day d. Why can she not have the diary a week later, on day $d + 7$?

b) [5] How often do B and D exchange the diary (in either direction) in the long run? Do H and I exchange less often?

c) [7] Suppose u has the diary, and v (a friend of u) desperately wants it. How long does v have to wait? Prove that the expected time for u to pass the diary to v is $1 + 2m_{u,v}(u)$ days, where $m_{u,v}(u)$ is the number of edges in u's connected component of the graph without edge (u, v). (Examples: $m_{B,D}(B) = 2$ and $m_{B,D}(D) = 5$.)

d) [5] Student u now has the diary on day d and writes a question for student $w \neq u$ into the diary (w is not necessarily a friend of u). What's the expected time before u can read an answer from w in the diary? Show that this expected time can only be one of these values: 16, 32, 48, or 64.

Model solution

a) The Markov chain has a period of 2, so if Alice has the book on day d, then she can only have the book on days $d + n$ where n is even.

b) The graph is connected (and finite) so the random walk is a (finite) irreducible MC, so it has a stationary distribution π. Let u, v be two friends. Then crossing the edge has probability

$$\pi_u \frac{1}{\delta(u)} + \pi_v \frac{1}{\delta(v)} = \frac{1}{2m} + \frac{1}{2m} = \frac{1}{m} = \frac{1}{8}$$

in the long run. (Note that this gives a uniform distribution over all the edges). Therefore the expected time between interactions of any friends u and v is $m = 8$.

c) Note that we can view the connected component (CC) containing u plus the edge (u, v) independently from the rest of the graph. This is because there is only one way to get from u's CC to v's CC, and that is through the edge (u, v). So all movements before crossing the edge (u, v) are independent of the rest of the graph (i.e. v's CC). Since v is a leaf in this graph we have a simple formula for $h_{v,v}$, that is, $h_{v,v} = 1 + h_{u,v}$. We know from the lecture that $h_{v,v} = \frac{2m}{\delta(v)}$, where $m = m_{u,v}(u) + 1$ equals the number of edges in this component, and $\delta(v) = 1$. Therefore

$$h_{u,v} = h_{v,v} - 1 = 2(m_{u,v}(u) + 1) - 1 = 2m_{u,v}(u) + 1.$$

d) For two friends, we have $c_{u,w} = h_{u,w} + h_{w,u} = 2m_{u,v}(u) + 1 + 2m_{u,v}(v) + 1 = 2m = 16$. On the other hand for any two nodes in a tree, there is unique path between them. For example to get from A to H, one has to reach B from A first and then D from B and H from D. So we can write $h_{A,H} = h_{A,B} + h_{B,D} + h_{D,H}$. Therefore

$$c_{A,H} = (h_{A,B} + h_{B,A}) + (h_{B,D} + h_{D,B}) + (h_{D,H} + h_{H,D}) = 3(16) = 48.$$

Indeed for any u, w at distance $d(u, w)$, we have $c_u, w = 16d(u, w)$. Since the diameter of the graph is 4, we have $c_{u,w} \in \{16, 32, 48, 64\}$ as required.

5 Elevator (20 points)

Consider an office building with 101 floors (numbered from 0 to 100) and 1 elevator that moves at a speed of 1 floor per second. Due to a pandemic only very few people are in the office and want to use the elevator. In particular, we can assume that there is always enough time between consecutive requests to move the elevator to any arbitrary floor. An elevator cannot wait between 2 floors. We would like to design an algorithm that decides where to position the elevator between requests.

The employees can request the elevator by pressing a button on any floor. The elevator then moves from its position to the employee and takes the employee to the floor they would like to go (which is different from the current floor). Opening and closing the doors as well as entering and exiting the elevator takes no time.

For each request, we consider the time between the employee pressing the button and the employee arriving at the desired floor. The cost of an algorithm is the sum of this time over all requests. The optimal (offline) algorithm knows all requests in advance.

a) [6] What is the best deterministic algorithm? (Design an algorithm, prove its competitive ratio, and argue that no deterministic algorithm with lower competitive ratio exists.)

b) [3] How would the optimum deterministic strategy change for 2 elevators? (One sentence answer is enough, no need to explain.)

Let RANDOM be the algorithm that moves the elevator to one of the floors uniformly at random after each request. Assume that the adversary knows the algorithm but not the individual random choices of RANDOM.

c) [6] Compute the competitive ratio of RANDOM.

d) [5] Prove that there exists no randomized algorithm with a lower competitive ratio than RANDOM.

Model solution

a) The optimal strategy is to always let the elevator wait at floor 50. Then every request for travelling f floors can be served in at most $50 + f$ seconds while OPT can serve it in f. Since $f \geqslant 1$, this means the algorithm is 51-competitive.

No deterministic algorithm can have a smaller comp. ratio than 51 since for any floor the elevator waits on, the adversary can request a trip of 1 floor starting 50 floors away.

b) Let the elevators wait on floors 25 and 75.

c) To maximize the time it takes the elevator to get there, the adversary will request trips starting on floor 0 or 100. Then the elevator will take

$$\sum_{i=0}^{100} \frac{1}{101} i = \frac{1}{101} \frac{100 \cdot 101}{2} = 50$$

seconds in expectation to arrive. So a request for moving f floors is served in $50 + f$ by RANDOM and in f by OPT. Thus, the adversary will choose $f = 1$ and RANDOMs comp. ratio is 51.

d) We use the Von Neumann/Yao Principle: Consider the distribution where trips from floor 0 to 1 and from floor 100 to 99 are requested with 50% probability each. No matter on which floor w the elevator waits, it will always need

$$\frac{1}{2} w + \frac{1}{2}(100 - w) + 1 = 51$$

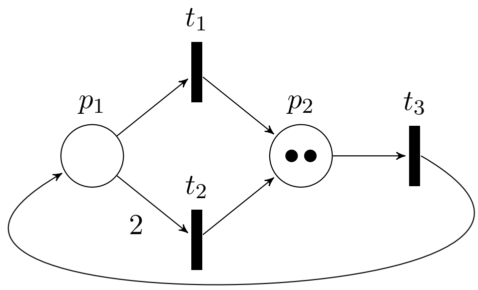
seconds in expectation to fulfill each request.

6 True or False (6 points)

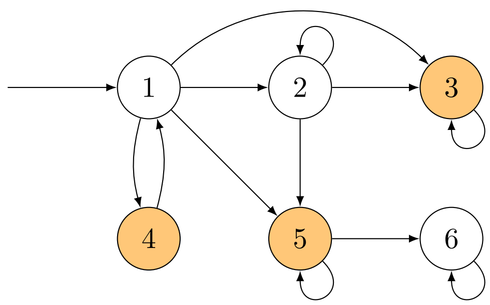
For each of the following statements, assess if it is true or false and tick the corresponding box. No justification is needed.
Every correct answer grants one point. Leaving a statement blank gives 0 point. Every incorrect answer looses one point on the total, with a minimum of 0 point for the whole question.

Notes:
- Petri net **P** is depicted in Figure 2(a).
  Automaton **A** is depicted in Figure 2(b).
- $[\![\, p \,]\!]$ denotes the set of states which satisfies property p.
  For example for automaton **A**, $[\![\, p \,]\!] = \{\, 3,\ 4,\ 5 \,\}$.

	Statement	**True**	**False**
1	The Binary Decision Diagram of a Boolean function is always unique.		
2	A non-timed Petri net can always be modeled by a deterministic finite automaton.		
3	Two Petri nets have different past markings, but the same current marking. They have different current states.		
4	The given Petri net **P** is deadlock-free.		
5	For automaton **A**, $[\![\ \mathrm{EF}\ (\ (\mathrm{EX}\ \overline{p})\ \texttt{AND}\ (\mathrm{AG}\ p)\ )\ ]\!] = \{1, 2\}$.		
6	Automaton **A** satisfies EG $\overline{p}$.		



(a) Petri net **P**



(b) Automaton **A** – 1 is the initial state. Property p is true only in states 3, 4, and 5.

Figure 2: Petri net **P** (2(a)) and Automaton **A** (2(b))

Model solution

	Statement	True	False
1	The Binary Decision Diagram of a Boolean function is always unique.		X
2	A non-timed Petri net can always be modeled by a deterministic finite automaton.		X
3	Two Petri nets have different past markings, but the same current marking. They have different current states.		X
4	The given Petri net **P** is deadlock-free.	X	
5	For automaton **A**, $[\![$ EF ((EX $\overline{p}$) `AND` (AG p)) $]\!] = \{1, 2\}$.		X
6	Automaton **A** satisfies EG $\overline{p}$.	X	

7 Binary Decision Diagram (10 points)

a) [6] Given the Boolean expression of function f and the ordering of variables $x_1 \prec x_2 \prec x_3 \prec x_4$, construct the reduced ordered binary decision diagram (ROBDD) of f. Merge all equivalent nodes, including the leaves.
Note: Use solid lines for *True* arcs and dashed lines for *False* arcs.

$$f(x_1, x_2, x_3, x_4) = x_1 \cdot x_2 \cdot (\,\overline{x_2} \cdot x_3 + \overline{x_3}\,) + \overline{x_3 \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4}$$

b) [2] Consider the BDD of the function g in Figure 3. Express g as a boolean function.

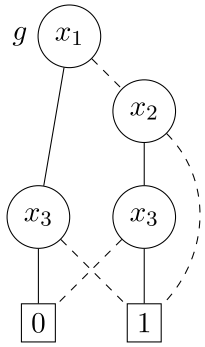
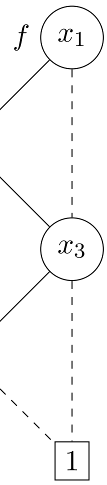


Figure 3: BDD of the Boolean function g

c) [2] Simplify the BDD of g (Figure 3) when $x_3 = 0$.

Model solution

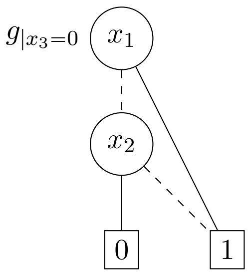
a) The BDD of f is shown below;



b) The function g can be expressed by the following boolean expression

$$g(x_1, x_2, x_3) = x_1 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot x_2 \cdot x_3$$

c) Finally, the BDD of g when x_2 evaluated to 0 is shown below;



8 Petri nets (24 points)

This question contains 3 independent sub-questions related to Petri nets. Throughout this question, we use the following notations.

- $M^\top = [m_1, m_2, m_3, m_4]$ and $U^\top = [u_1, u_2, u_3, u_4, u_5]$ are marking and firing vectors of **P**, respectively.

- m_i denotes the number of tokens in place p_i.

- u_i denotes the number of firings of transition t_i.

Let us first consider the Petri net **P1** in Figure 4.

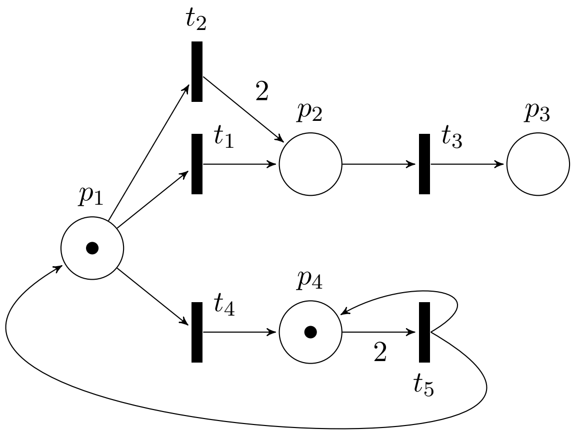


Figure 4: Petri net **P1** – Circles, dots and bars represent places, tokens and transitions, respectively. Weights are associated to edges when they are different from 1.

8.1 Reachability [8 points]

a) [2] Derive the incidence matrix A of the Petri net **P1** from Figure 4.

b) [2] Given initial marking $M_0^\top = [1, 0, 0, 1]$, is the Petri net deadlock-free? If so, provide a brief proof. If not, list all reachable markings where the network is in deadlock.

c) [2] Consider the firing vector $U_S^\top = [0, 1, 1, 1, 1]$, where S denotes a firing sequence containing the firing of t_2, t_3, t_4 and t_5 once.
Use the incidence matrix and the state equation of the Petri net **P1** to compute the marking $M_1^\top$ obtained from the initial marking $M_0^\top = [1, 0, 0, 1]$ after firing S.

d) [2] Given initial marking $M_0^\top = [1, 0, 0, 1]$ and assume $M_1^\top$ computed in (c) is a valid marking (i.e., all elements are non-negative). Is it true that any firing sequence S with firing vector $U_S^\top = [0, 1, 1, 1, 1]$ is feasible? Provide a brief proof of your conclusion.

Model solution

a) $A = \begin{bmatrix} -1 & -1 & 0 & -1 & 1 \\ 1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$

b) Deadlocks with $[0, 0, 2, 1]$ or $[0, 0, 1, 1]$

c)

$$M_1 = A \cdot T_S + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

d) Not necessary a valid firing sequence. If t_2 fires first, t_5 can never fire.

8.2 Coverability [8 points]

Let us now consider the Petri net **P2** in Figure 5.

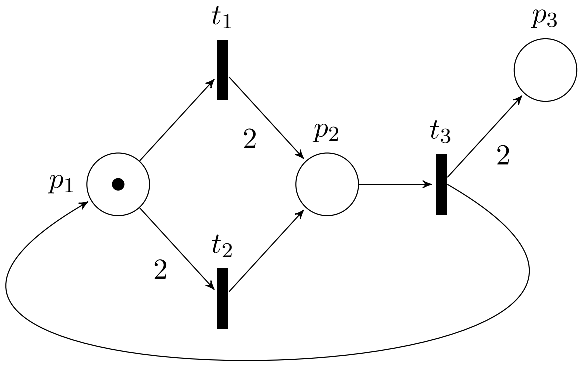


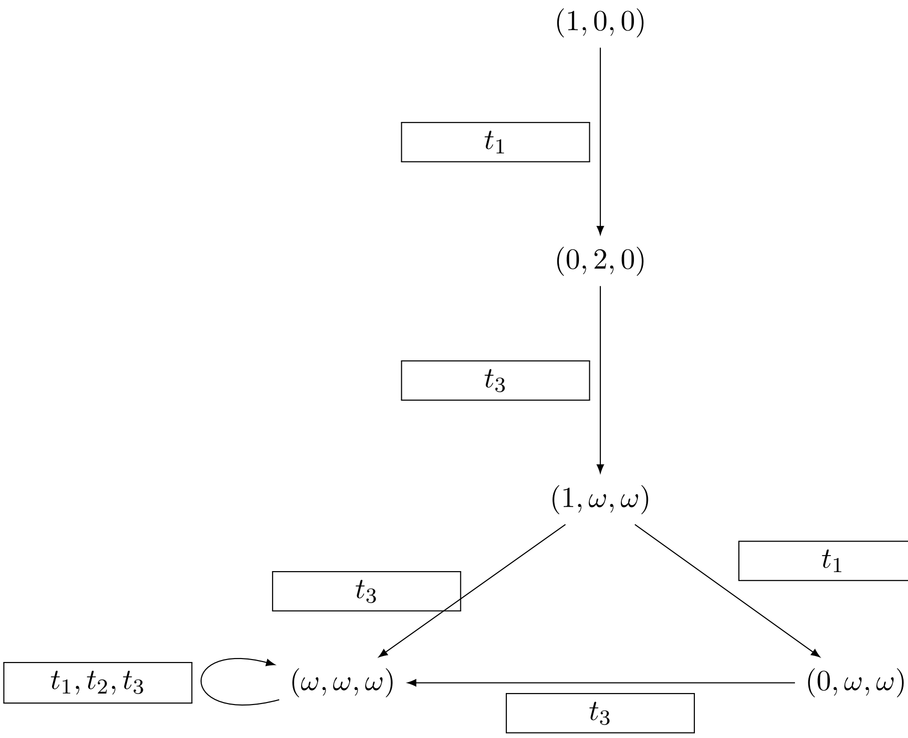
Figure 5: Petri net **P2** – Circles, dots and bars represent places, tokens and transitions, respectively. Edge weights are marked close to the edge when they are different from 1.

a) [4] Construct the coverability graph of the Petri net **P2**. **Note:** The coverability graph is obtained from the coverability tree by merging nodes with the same marking.

b) [3] Is $M^\top = [7, 10, 2]$ reachable? Justify.

c) [1] Is $M^\top = [2, 2, 9]$ reachable? Justify.

a)

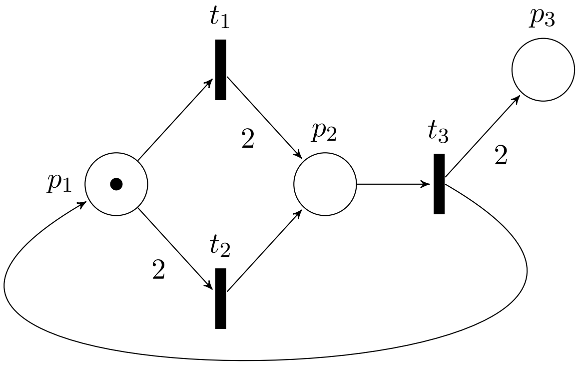


b) No, $M^t = [7, 10, 2]$ is not reachable from $M_0^t = [1, 0, 0]$. Because t_3 is the only predecessor of p_1 and p_3, and p_3 has no out-going connection, we have $M(p_3) \geqslant M(p_1) - 1$. Hence $M^t = [7, 10, 2]$ is not reachable.

c) No, p_3 can only have even number of tokens.

8.3 Timed Petri Net `[8 points]`

The Petri net **P2** is redrawn below:



But now the transitions are associated with **delays** between their activation and firing:

$$d(t_1) = 2$$
$$d(t_2) = 1$$
$$d(t_3) = 2$$

Below is a table for the simulated steps, with columns for number of steps, simulation time τ, firing vector T^τ, current state M^τ, and event list L^τ. Some rows or cells are already given. Simulate the behaviour of the timed Petri net **P2** and fill in the table.

Notes:

- If there are several transitions enabled at the same time, they fire in the ordering of their index, i.e., the smaller index fires first.
- Every firing is a simulated step.

steps	τ	T^τ	M^τ	L^τ
0	0	-	$[1,0,0]$	$(t_1,2)$
1	2	$[1,0,0]$	$[0,2,0]$	$(t_3,4)$
2				
3				
4				

Model solution

steps	τ	T^τ	M^τ	L^τ
0	0	-	$[1, 0, 0]$	$(t_1, 2)$
1	2	$[1, 0, 0]$	$[0, 2, 0]$	$(t_3, 4)$
2	4	$[0, 0, 1]$	$[1, 1, 2]$	$(t_1, 6), (t_3, 6)$
3	6	$[1, 0, 0]$	$[0, 3, 2]$	$(t_3, 6)$
4	6	$[0, 0, 1]$	$[1, 2, 4]$	$(t_1, 8), (t_3, 8)$